TD 5 – Non-déterminisme et classe NP

Exercice 1. . Un autre point de départ NP-complet

1. Expliquer le fonctionnement d'une machine de Turing non déterministe fonctionnant en temps polynomial qui reconnaisse $L_1 = \{(\langle M \rangle, t) : t \leqslant |Q| \text{ et } M \text{ admet un calcul acceptant sur le mot vide en temps } \leqslant t\}$.
2. Montrer que L_1 est NP-complet. On pourra commencer, étant donnée une machine de Turing non déterministe M fonctionnant en temps polynomial $p(n)$ et un mot x, construire une machine de Turing M_x avec un nombre d'états $\geqslant 2|x| + p(|x|)$ qui admet un calcul acceptant sur le mot vide en temps $2|x| + p(|x|)$ ssi M admet un calcul acceptant sur x en temps $\leqslant p(|x|)$.
3. On considère un modèle de machine de Turing non déterministe avec un seul ruban en lecture-écriture. Montrer que si M est non déterministe au sens du cours, on peut trouver une machine M' avec un seul ruban qui reconnait le même langage que M et qui fonctionne en temps $\leqslant K(1 + t^2)$ où t est le temps de calcul de M et K est une constante.
4. En déduire que le langage $L_1' = \{(\langle M \rangle, t) : t \leqslant |Q|, M \text{ a un seul ruban et } M \text{ admet un calcul acceptant sur le mot vide en temps } \leqslant t\}$ est NP-complet.
5. On considère maintenant le problème du pavage fini du plan PFP suivant :
   — **Donnée** : Un ensemble fini C de couleurs, dont une couleur *blanche*, un ensemble fini T de tuiles carrées 1×1 dont les bords sont colorés par des éléments de C, un entier $n \leqslant |C|$.
   — **Question** : Peut-on trouver un coloriage du carré $n \times n$ par des tuiles dont les côtés adjacents sont de la même couleur, avec toutes les couleurs extérieures blanches ?
   Montrer que PFP$\in$NP.
6. Montrer que PFP est NP-complet en y réduisant polynomialement le langage L_1'. On pourra construire un pavage réprésentant l'évolution d'une machine de Turing non déterministe sur un calcul terminant.
7. En déduire que Sat est NP-complet.

Exercice 2. Montrer que la classe E n'est pas close pour $\leqslant_m^p$.

Solution de l'exercice 2. On commence par prendre dans EXP un langage L qui n'est pas dans E, puis on le modifie par padding en un langage L' dans E, avec $L \leqslant L'$, ce qui donne le contre-exemple souhaité.

Exercice 3. Montrer que le problème $A = \{(\langle M \rangle, x, t) \mid M(x) \text{ accepte en temps } t\}$, où M est une machine non-déterministe, x est un mot et t un entier en binaire, est NEXP-complet.

Solution de l'exercice 3. Comme dans le livre de Perifel pour NP.

Exercice 4. Si P= NP, montrer que le problème de calculer une affectation de valeurs satisfaisant une formule propositionnelle peut être résolu en temps polynomial.

Solution de l'exercice 4. Pour trouver une affectation de valeurs à une formule $\psi(x_1, ..., x_n)$, on peut d'abord tester si elle est satisfaisable. Si elle l'est, alors on teste si $\psi(0, x_2, ..., x_n)$ est satisfaisable, et on affecte alors la valeur 0 à x_2, et sinon on affecte la valeur 1. on recommence ainsi de suite pour affecter une valeur à chaque variable.

Exercice 5. On rappelle que INDSET est l'ensemble des couples (G, k) avec G graphe non dirigé et $k \in \mathbb{N}$ tel que G possède un sous-ensemble indépendant de taille k (c'est-à-dire un ensemble de k sommets jamais connectés deux à deux). On définit HALF-INDSET l'ensemble des graphes $G = (V, E)$ non dirigés possédant un sous-ensemble indépendant de taille $\lceil |V|/2 \rceil$.

1. Montrer que HALF-INDSET $\in \mathsf{NP}$.
2. Montrer que INDSET $\leqslant_m^p$ HALF-INDSET.
3. En déduire que le problème HALF-INDSET est NP-complet.

Solution de l'exercice 5.

1. Il suffit de deviner un sous-ensemble des sommets et de vérifier qu'il a la bonne taille et est un sous-ensemble indépendant, ce qui se fait en temps polynomial.
2. On va réduire INDSET en distinguant trois cas :
   — Si l'instance (G, k) est telle que k vaut exactement $\lceil |V|/2 \rceil$, on renvoie G.
   — Si $k < \lceil |V|/2 \rceil$, on ajoute assez de nouveaux sommets indépendants pour que tout sous-ensemble indépendant de taille au moins k dans G devienne un sous-ensemble indépendant de taille au moins la moitié dans le nouveau graphe.
   — Si $k > \lceil |V|/2 \rceil$, on ajoute de nouveaux sommets connectés entre-eux et connectés à tous les sommets de G, pour de nouveau avoir la même propriété.
3. Évident.

Exercice 6. Colorabilité.

1. Montrer que le problème de k-COLORABILITÉ se réduit polynomialement au problème de $(k+1)$-COLORABILITÉ.
2. Dans quelle classe de complexité est le problème 2-COLORABILITÉ ?
3. Montrer que le problème 3-COLORABILITÉ est NP-complet.

Solution de l'exercice 6.

1. Soit $f : G \mapsto G' = (V', E')$ avec $V' = V \cup \{u\}$ où $u \notin V$, et avec $E' = E \cup \{(u, v) \mid v \in V\}$. (On rajoute un sommet relié à tous les autres, on doit donc nécessairement le colorier d'une couleur différente.) G est k-coloriable ssi $f(G) = G'$ est $(k+1)$-coloriable, donc k-COLORABILITÉ se réduit polynomialement à $(k+1)$-COLORABILITÉ.
2. Montrons que 2-COLORABILITÉ $\in \mathsf{P}$. Décrivons un algorithme qui sur la donnée d'un graphe $G = (V, E)$ vérifie en temps polynomial si G est 2-coloriable.
   Le graphe G est 2-coloriable ssi chaque composante connexe de G l'est. Pour G connexe, on peut faire un parcours de graphe (on a vu dans un exercice précédent qu'on peut bien faire ceci en temps polynomial). On peut supposer que le premier sommet est colorié avec la couleur 1, et le parcours permet de colorier tous les sommets avec les couleurs 1 et 2 de manière nécessaire : effectivement, lorsqu'on découvre un sommet v à partir d'un sommet u, v doit nécessairement être d'une couleur différente de celle de u (donc de l'autre couleur, on n'a pas le choix). Cette étape se fait en temps polynomial. On peut ensuite tester en temps polynomial si ce coloriage est propre (deux sommets reliés sont toujours de couleurs distinctes).

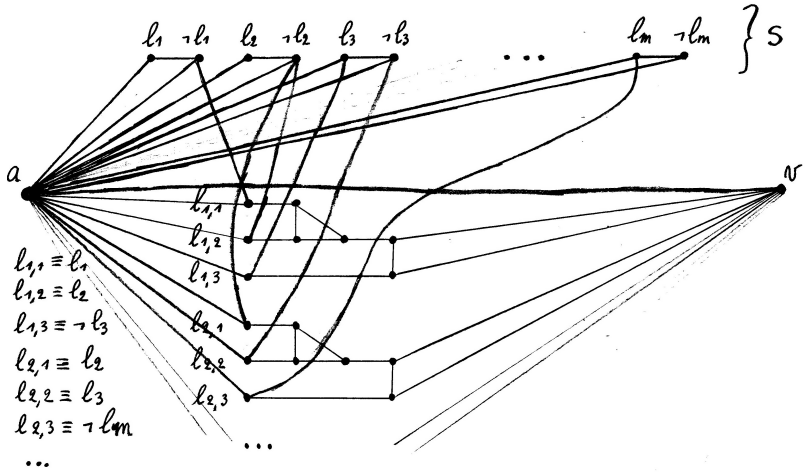
   Autre méthode. On a montré que 2SAT $\in \mathsf{P}$. Il est très facile de réduire 2-COLORABILITÉ à 2SAT en temps polynomial. Soit $G = (V, E)$ avec $V = \{1, \ldots, n\}$. Le graphe G est 2-coloriable ssi la formule $F = \bigwedge_{(i,j)\in E}((x_i \vee \neg x_j) \wedge (x_j \vee \neg x_i))$ est satisfaisable. Ceci montre que 2-COLORABILITÉ $\in \mathsf{P}$.
3. On va montrer que le langage 3-COLORABILITÉ est NP-complet.

— 3-COLORABILITÉ est dans NP.
Sur la donnée d'un graphe $G = (V, E)$ et d'une coloration du graphe en trois couleurs, on peut vérifier en temps polynomial si la coloration est telle que deux sommets liés sont de couleurs différentes.

— 3-COLORABILITÉ est NP-dur.
On va réduire 3SAT à 3-COLORABILITÉ. Étant donné une formule sous forme 3CNF, on construit en temps polynomial un graphe qui est 3-coloriable ssi la formule est satisfaisable.
Les trois couleurs qu'on utilise sont vrai (V), faux (F) et autre (A). Maintenant on cherche un petit sous-graphe pour représenter l'opération booléenne 'OU'. Si a et b sont colorié avec V ou F, il existe un coloriage qui donne la valeur V à c si et seulement si au moins l'un des deux sommets a ou b est colorié V, sinon c doit être colorié F.

Soit $\phi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$ une formule $3CNF$, avec les clauses $\varphi_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$.
Pour le graphe G_φ correspondant on a :
- Un sommet a.
- Un ensemble S de sommets avec un sommet pour chaque littéral de ϕ, et un sommet pour chaque négation (pour contrôler les valeurs donné à chaque littéral et sa négation). Tous liés au sommet a.
- Un graphe de clause pour chaque clause φ_i.
- Chaque littéral d'un graphe de clause est lié au sommet a.
- Chaque littéral d'un graphe de clause et lié à sa négation dans S.
- Les sorties de chaque graphe de clause sont le même sommet v qui est aussi lié à a.

Cette transformation peut se faire en temps polynomial.



On doit montrer que $\phi \in$ 3SAT si et seulement si G_ϕ est 3-coloriable.
[⇒]
Soit $\phi \in$ 3SAT, alors ϕ est satisfaisable par une affectation σ des variables. On définit un coloriage de G_ϕ où un sommet l_1 a la valeur V si $\sigma(l_1) = V$ et la couleur F sinon. Comme σ est une affectation, le coloriage est correct pour l'instant. On donne la couleur A au sommet a. Car ϕ est satisfaisable, on peut colorier chaque graphe de clause pour que le sommet v de sortie (commun) ait la couleur V et que deux sommets liés soient de différentes couleurs.
[⇐]
Si G_ϕ est coloriable avec trois couleurs, montre que ϕ est satisfaisable. Soit A la couleur de a, soit V la couleur de v.
- Chaque littéral (sommet) a une couleur V ou F (F est la troisième couleur)
- Si on a l et $\neg l$ dans le graphe, ils ne peuvent pas avoir la même couleur.

- Soit φ_i une clause de ϕ, le graphe de clause correspondant a des couleurs V ou F en entrée et V en sortie : donc au moins une des entrées est coloriée V.
- On définit $\sigma : \sigma(u) = V$ si et seulement si le littéral u est coloriée V ou si le littéral $\neg u$ est colorié F.
- σ est bien défini.
- σ satisfait ϕ.

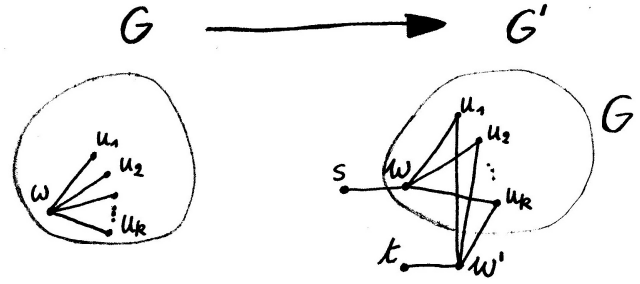
Exercice 7. Cycles.

1. Montrer que le problème EC (l'ensemble des graphes ayant un cycle eulérien) est dans P.
2. Montrer que les problèmes UHC (graphes ayant un cycle hamiltonien) et UHP (graphes ayant un chemin hamiltonien) sont polynomialement équivalents.
3. Montrer que les problèmes DHC (graphes orientés ayant un cycle hamiltonien) et DHP (graphes orientés ayant un chemin hamiltonien) sont polynomialement équivalents.
4. Montrer que les problèmes DHC (graphes orientés ayant un cycle hamiltonien) et UHC (graphes ayant un cycle hamiltonien) sont polynomialement équivalents.
5. Sachant que DHP est NP-complet. En déduire que UHC, DHP, DHC sont également NP-complets.

Solution de l'exercice 7.

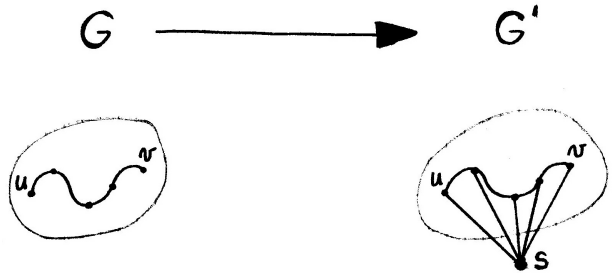
1. On montre qu'un graphe G connexe a un cycle eulérien si et seulement si chaque sommet est de degré pair.
   [$\Rightarrow$]
   Soit C un cycle eulérien. Si on regarde un sommet x quelconque, alors, suivant le cycle C, pour chaque arête qui 'arrive' dans x, il existe un arête qui 'part' de x. Un cycle eulérien parcourt chaque arête, alors chaque sommet a un nombre pair d'arêtes, donc est de degré pair.
   [$\Leftarrow$]
   Soit C un cycle de longueur maximale de G. Supposons qu'il ne passe pas par toutes les arêtes de G : il existe une arête (u, u') qui n'est pas dans C. Comme G est connexe, on peut en plus trouver une arête de $(v, v') \in G \setminus C$ adjacente à C (considérer un plus cours chemin de u à C si aucun des sommets u et u' ne se trouve sur C). Il existe donc une arête (v, v') adjacente à C et qui n'est pas dans C. En partant de cette arête et en se déplaçant dans $G \setminus C$ (dont chaque sommet est de degré pair), on retombe nécessairement sur le sommet v à une certaine étape, ce qui fournit un cycle C' disjoint de C (au niveau des arêtes) intersectant C en v (et peut-être en d'autres sommets). On peut alors fusionner les deux cycles C et C' pour en obtenir un plus grand. Il était donc absurde de supposer que C ne passait pas par toutes les arêtes : C est bien un cycle Eulérien.
   Donc contrôler si un graphe a un cycle eulérien consiste à déterminer si :
   - G est connexe.
   - G est différent d'un seul sommet.
   - Chaque sommet de G est de degré pair. Pour regarder si chaque sommet d'un graphe est de degré pair, on regarde son matrice d'adjacence. Pour chaque ligne on fait la somme de ses nombres, les différentes sommes doivent toujours être paires. On peut faire ces sommes en temps polynomial.
2. Pour deux ensembles A et B on a $A \equiv_p B$ si et seulement si $A \leqslant_p B$ et $B \leqslant_p A$.
   — Pour UHC $\leqslant_p$ UHP : il faut donner une transformation en temps polynomial $G \to G'$ telle que G a un cycle hamiltonien si et seulement si G' a un chemin hamiltonien.

Soit G un graphe. Si G n'a pas de sommet de degré $\geqslant 2$, on renvoie le graphe constitué de deux sommets sans arêtes. Sinon, soit w un sommet de degré au moins 2 de G. Pour G', on ajoute un sommet w' relié à tous les voisins de w. On ajoute un sommet s relié à w et un sommet t relié à w'.



[⇒]
Si G a un cycle hamiltonien qui passe par w, on le voit comme partant de w et de revenant à w. On en déduit un chemin hamiltonien qui part de s, va à w, utilise le cycle pour passer tous les sommets différent de w, pour aller à w', et finir à t.
[⇐]
Si G' a un chemin hamiltonien, on peut le voir comme allant de s à t : il donne donc un chemin hamiltonien de w à w', ce qui donne un cycle hamiltonien dans G (de w à w).

— Pour UHP $\leqslant_p$ UHC : il faut donner une transformation en temps polynomial $G \to G'$ telle que G a un chemin hamiltonien si et seulement si G' a un cycle hamiltonien.
Soit G un graphe. Si G est réduit à un seul sommet, $G' = G$. Sinon, pour construire le graphe G', on ajoute à G un nouveau sommet s relié à chaque sommet de G.



[⇒]
Si G a un chemin hamiltonien, on peut le voir comme allant de u à v : il donne donc un cycle hamiltonien de s à u à v à s dans G'.
[⇐]
Si G' a un cycle hamiltonien qui passe par s, on le voit comme partant de s et de revenant à s. On en déduit un chemin hamiltonien dans G qui part du sommet u, qui est après s dans le cycle, utilise le cycle pour passer tous les sommets différent pour aller à v, le précédent de s dans le cycle.

3. Similaire au cas non-orienté.
4. Montrons que UHC $\equiv_p$ DHC.
   — UHC $\leqslant_p$ DHC.
   Soit $G = (V, E)$ un graphe non-orienté. Soit $G' = (V, E')$ où $E' = \{(u, v), (v, u) \mid (u, v) \in E\}$. Il est immédiat que G' a un circuit Hamiltonien ssi G a un cycle Hamiltonien.
   — DHC $\leqslant_p$ UHC.
   Soit $G = (V, E)$ un graphe orienté. On pose $V' = \{u, u^+, u^- \mid u \in V\}$ et $E' = \{(u^+, v^-) \mid (u, v) \in E\}$. Soit $G' = (V', E')$. Clairement, on peut calculer G' en temps polynomial à partir de G. Montrons que G' a un cycle Hamiltonien ssi G a un circuit Hamiltonien.
   [⇒]

Si G possède le circuit Hamiltonien $u_1 \to u_2 \to \ldots \to u_k \to u_1$, alors $u_1^- \to u_1 \to u_1^+ \to u_2^- \to \ldots \to u_k \to u_k^+ \to u_1^-$ est un cycle Hamiltonien de G'.
[$\Leftarrow$]
Supposons que G' a un cycle Hamiltonien. Soit u_1 un sommet quelconque de V' qui n'est pas de type '+' ou '-'. On peut parcourir le cycle dans n'importe laquelle des deux directions à partir de u_1, parcourons le dans le sens $u_1 \to u_1^+ \to \ldots$. Donc nécessairement, ce chemin commence par $u_1 \to u_1^+ \to u_2^-$. Le sommet suivant est soit u_2, soit du type u_3^+ : montrons que cette deuxième option n'est pas possible.
Si le chemin est du type $u_1 \to u_1^+ \to u_2^- \to u_3^+ \to \ldots$, il doit nécessairement passer par u_2. Pour passer par u_2, il doit nécessairement venir de u_2^+. Mais le chemin ne peut plus quitter u_2 sans repasser par un sommet déjà visité.
Le cycle Hamiltonien de G' commence donc par $u_1 \to u_1^+ \to u_2^- \to u_2 \to u_2^+$ et comme le raisonnement précédent s'applique à chaque étape, il est de la forme $u_1 \to u_1^+ \to u_2^- \to \ldots \to u_k \to u_k^+ \to u_1^- \to u_1$. Mais alors $u_1 \to u_2 \to \ldots \to u_k \to u_1$ est un circuit de G.

5. On a vu en cours que DHP est NP-complet. Par réduction, les autres problèmes sont NP-complets également.

Exercice 8. Satisfaisabilité d'un circuit booléen.
Un circuit booléen est un graphe $C = (V, E)$ où les sommets dans $V = \{1, 2, \ldots, n\}$ sont appelés les portes du circuit. Ce graphe est orienté, sans cycle, et on peut donc supposer que les arêtes sont de la forme (i, j) avec $i < j$. Chaque sommet du graphe a un degré entrant égal à 0, 1 ou 2 et est étiqueté par l'un des éléments de $\{0, 1, \neg, \wedge, \vee\} \cup \{x_i \mid i \in \mathbb{N}^*\}$. Les sommets étiquetés par $0, 1$ ou une variable sont de degré entrant 0, ceux étiquetés par $\neg$ de degré entrant 1, et ceux étiquetés par $\wedge$ ou $\vee$ de degré entrant 2. Le sommet numéroté n n'a pas d'arêtes sortantes. Il est facile de définir la valeur d'un circuit booléen pour une affectation des variables. On peut alors considérer le problème CIRCUIT-SAT : étant donné un circuit booléen, existe-t-il une affectation lui donnant la valeur "vrai"? Montrer que le problème CIRCUIT-SAT est polynomialement équivalent au problème SAT.

Solution de l'exercice 8. SAT se réduit facilement à CIRCUIT-SAT car une formule est un cas particulier de circuit. Il est facile de voir que CIRCUIT-SAT est dans NP, ce qui implique l'existence d'une réduction dans l'autre sens, mais nous donnons ici une réduction directe.
Soit C un circuit de taille n on introduit de nouvelles variables $y_1, \ldots, y_n$, une pour chaque porte du circuit. Pour chaque porte nous exprimons par des clauses que la porte prend la bonne valeur : si i est une porte étiquetée par 0 ou 1 nous mettons $\neg y_i$ ou y_i respectivement ; si i est étiquetée par une variable x_j nous mettons $y_i \Leftrightarrow x_j$ ce qui s'exprime par les deux clauses $y_i \vee \neg x_j$ et $\neg y_i \vee x_j$; si i est étiquetée par une négation avec une flèche provenant de la porte j nous mettons $y_i \Leftrightarrow \neg y_j$, ce qui s'exprime par les deux clauses $\neg y_i \vee \neg y_j$ et $y_j \vee y_i$; si la porte i est étiquetée par $\vee$ avec des flèches provenant des portes j et k, nous mettons $y_i \Leftrightarrow y_j \vee y_k$, ce qui s'exprime par les trois clauses $\neg y_i \vee y_j \vee y_k$ et $y_i \vee \neg y_j$ et $y_i \vee \neg y_k$; le cas d'une porte $\wedge$ se traite de manière similaire. Le circuit C et la formule obtenue par conjonction de ces clauses sont alors équi-satisfaisables.

Exercice 9. Couverture d'un graphe par des sommets.
Soit $G = (V, E)$ un graphe et S un sous-ensemble de V. On dit que S est une couverture de sommets de G si, pour toute arête (u, v) de E, au moins une des ses extrémités (u et v) appartient à S. On appelle VERTEXCOVER l'ensemble des couples (G, k) tels que G est un graphe ayant une couverture de sommets de taille au plus k.

1. Si $G = (V, E)$ est un graphe et S une couverture de sommets de G, que pouvez-vous

dire de l'ensemble de sommets $V \setminus S$ dans G ?

2. Montrer que VertexCover est NP-complet.
3. On appelle VertexCoverEven l'ensemble des (codages de) couples (G, k) tels que G est un graphe dont tous les sommets sont de degré pair et ayant une couverture de sommets de taille inférieure ou égale à k. Montrer que VertexCoverEven est NP-complet.

Solution de l'exercice 9.

1. Montrons que $V \setminus S$ est un sous-ensemble indépendant de G : Si $u, v \in V \setminus S$, alors il ne peut y avoir d'arêtes entre u et v car sinon celle-ci ne serait pas couverte par la couverture S.
2. VertexCover est dans NP, parce que la question de l'appartenance d'un couple (G, k) revient à une question d'existence d'une solution et parce sur la donnée d'un couple (G, k) et d'un ensemble S candidat à être solution on peut vérifier en temps polynomial si S est bien une couverture des sommets de taille au plus k.
   VertexCover est NP-complet par réduction de IndSet, problème NP-complet vu en cours : on considère la réduction qui à $(G = (V, E), k)$ associe $(G, |V| - k)$; cette réduction est clairement calculable en temps polynomial. On a vu à la question 1 que si $S \subset V$ est une couverture des sommets de G, alors $V \setminus S$ est un ensemble de G. Il est immédiat que la réciproque est vraie également. Ainsi, un graphe sur n sommets possède un sous-ensemble indépendant de taille k ssi il possède une couverture de sommets de taille $n - k$.
3. On va réduire VertexCover à VertexCoverEven : cela veut dire transformer un couple (G, k) en un nouveau couple (G', k') tel que (G, k) a une couverture de taille k ssi G' a une couverture de taille k' et tel que dans G' tous les sommets soient de degré pair. Une façon de faire est de considérer tous les sommets de G de degré impair. On peut montrer qu'il y a forcément un nombre pair de tels sommets (utiliser que la somme des degrés des sommet d'un graphe est deux fois le nombre de ses arêtes, donc pair). Pour obtenir G' on ajoute un nouveau sommet u relié à ces derniers. Tous les sommets de G' sont alors bien de degré pair, et si G a une couverture de taille k, alors G' a une couverture de taille $k + 1$ en ajoutant le sommet u. Par contre il n'est pas clair que si G' a une couverture de taille $k + 1$ alors G a forcément une couverture de taille k, car la couverture de taille $k + 1$ de G' pourrait ne pas faire intervenir u. Pour remédier à cela on ajoute à G' deux sommets v, w et les arêtes pour faire un triangle u, v, w. Comme précédemment, si G a une couverture de taille k alors G' en a une de taille $k + 2$ en ajoutant les sommets u et v par exemple. Dans l'autre sens, si G' a une couverture de taille $k + 2$, alors parmi les sommets u, v, w il y en a au moins 2 dans la couverture, donc les sommets qui restent sont une couverture de sommets de taille au plus k de G.
   On peut aussi dupliquer le graphe en G_1 et G_2 et relier en plus chaque somme de G_1 aux copies dans G_2 de ses voisins dans G_1, et réciproquement. Tous les sommets deviennent de degré pair et la taille d'une couverture de sommets double.

Exercice 10. Ensemble de sommets dominant un graphe.

On dit qu'un sous-ensemble D des sommets d'un graphe $G = (V, E)$ est dominant si tout sommet de G est soit dans D soit relié à un sommet de D par une arête. Le problème Dom est l'ensemble des couples (G, k) où G est un graphe possédant un sous-ensemble dominant de taille au plus k.

1. Soit $G = (V, E)$ un graphe. On construit un graphe $G' = (V', E')$ de la façon suivante : V' est l'ensemble V, moins les sommets isolés (n'appartenant à aucune arête),

et auquel on ajoute des nouveaux sommets u_e pour chaque arête $e \in E$; $E' = E \cup \{(x, u_e) \mid x \text{ est une extrémité de } e\}$. Montrer que G a une couverture de sommets de taille au plus k ssi G' a un ensemble dominant de taille au plus k.

2. En déduire que Dom est NP-complet.

Solution de l'exercice 10.

1. Supposons d'abord que G a une couverture de sommets C de taille au plus k. On peut supposer que C ne contient aucun sommet isolé (sinon on les enlève). Montrons que C est un ensemble dominant de G' :
   — Soit x un sommet de V non isolé dans G. Alors x est relié à au moins un autre sommet y. Comme C est une couverture des sommets, l'arête (x, y) est couverte, donc $x \in C$ ou $y \in C$. Dans les deux cas, le sommet x de G' est bien à distance au plus 1 d'un sommet de C.
   — Soit x un sommet de G' de type u_e. Alors x est relié aux deux extrémités de l'arête e et l'un de ces deux sommets appartient à C car C couvre l'arête e. Encore une fois, x est à distance au plus 1 d'un sommet de C.

   Réciproquement, supposons que G' a un ensemble dominant D de taille k. Soit e une arête de G. Si aucune de ses deux extrémités n'appartient à D, alors u_e doit appartenir à D, car sinon il ne serait ni dans D ni relié à un sommet dans D. On peut donc choisir l'une des extrémités de e et la mettre dans D et enlever u_e de D. En faisant ceci pour toutes les arêtes, on obtient une couverture de sommets de taille (au plus) k pour G.
2. Dom est donc NP-complet : il est dans NP, car sur la donnée d'un couple (G, k) et d'un ensemble D, on peut vérifier en temps polynomial si D est un ensemble dominant pour G de taille au plus k ; il est NP-dur par réduction depuis VertexCover, en appliquant la construction de la question précédente.

Exercice 11. Soit $G = (V, E)$ un graphe et S un sous-ensemble de V. On dit que S est une couverture de sommets de G si, pour toute arête (u, v) de E, au moins une des ses extrémités (u et v) appartient à S. On appelle VertexCover l'ensemble des couples (G, k) tels que G est un graphe ayant une couverture de sommets de taille inférieure ou égale à k. Un exercice de TD a montré que le problème VertexCover est NP-complet. On considère maintenant le problème Cover : sur la donnée d'un ensemble E ainsi que de parties $P_1, \ldots, P_m$ de E, et d'un entier k, déterminer s'il existe un sous-ensemble J de $\{1, \ldots, m\}$ tel que $|J| = k$ et l'union des P_i soit égale à E. Montrer que Cover est NP-complet.

Solution de l'exercice 11. Ce problème est clairement dans NP, car on peut deviner un tel sous-ensemble et vérifier en temps polynomial qu'il satisfait bien les conditions.
Pour montrer qu'il est NP-dur, faisons une réduction de VertexCover. Considérons une instance (G, k). On prend pour E l'ensemble des arêtes de G et on aura un ensemble P_u pour chaque sommet u de G, P_u contenant toutes les arêtes dont u est une extrémité. L'entier k reste inchangé. La transformation se fait bien en temps polynomial et il est clair qu'on a l'équivalence des appartenances des instances.

Exercice 12. Le problème SubsetSum.
Soit 3-Cover le problème suivant : sur la donnée d'un ensemble E ainsi que de parties $P_1, \ldots, P_k$ de E à 3 éléments chacune, déterminer s'il existe un sous-ensemble I de $\{1, \ldots, k\}$ tel que pour les indices dans I les P_i soient 2-à-2 disjoints et l'union des P_i soit égale à E.
Soit SubsetSum le problème suivant : sur la donnée d'entiers $a_1, \ldots, a_k, b$, déterminer s'il existe un sous-ensemble J de $\{1, \ldots, n\}$ tel que $\sum_{j \in J} a_j = b$. Sachant que 3-Cover est NP-complet, montrer que SubsetSum est NP-complet.

Solution de l'exercice 12. Il est facile de voir que SubsetSum est dans NP. Pour montrer que ce problème est NP-dur, nous allons faire une réduction de 3-Cover, comme l'énoncé nous le suggère.
On considère que l'ensemble E est l'ensemble $\{1, \ldots, n\}$. À chaque entier i dans E on associe un poids p^i pour un entier p fixé que l'on déterminera plus tard. À chaque partie $P_j = \{u, v, w\}$ de l'instance de 3-Cover que l'on souhaite transformer, on associe le poids $a_j = p^u + p^v + p^w$. On pose $b = \sum_{i=1}^{n} p^i$. Il est clair que si on a un sous-ensemble I de $\{1, \ldots, k\}$ tel que l'union des P_j correspondants soit disjointes et égale à E, alors ce choix de sous-ensemble est tel que $\sum_{j \in I} a_j = b$. Réciproquement, pourrait-on obtenir $b = \sum_{i=1}^{n} p^i$ autrement qu'en ayant "couvert" chaque élément de E ? Cela veut dire qu'on somme des entiers et qu'on obtient le nombre dont la décomposition en base p est donnée par le mot de longueur n constitué seulement de 1. Pour être sûr qu'on ne pourrait pas obtenir ce nombre autrement, il suffit d'être sur qu'on n'aura pas de problèmes de retenues, ce qui est possible si on prend $p > k$.

Exercice 13. Le voyageur de commerce.
On considère un graphe non-orienté complet (donc qui contient une arête entre toute paire de sommets distincts i et j) avec une matrice de distances D à coefficients entiers naturels : $D_{i,j}$ est la distance pour aller du sommet i au sommet j. On appelle TSP l'ensemble des (codages de) couples (D, k) tels qu'il existe un circuit passant exactement une fois par chaque sommet et tels que la distance totale parcourue (obtenue en additionnant les distances des arêtes empruntées) soit inférieure ou égale à k. Montrer que TSP est NP-complet.

Solution de l'exercice 13. Il est facile de voir que TSP est dans NP, car il s'agit bien d'un problème d'existence de solution et, sur la donnée d'une matrice de distances, d'un entier k et d'une suite de sommets on peut vérifier en temps polynomial si cette suite passe bien exactement une fois par chaque sommet et si la longueur totale est bien inférieure à k.
Pour montrer que TSP est NP-dur, nous allons réduire UHC à TSP. Soit donc G un graphe. Pour le transformer en une instance de TSP, il faut mettre des poids sur les arêtes et décider d'un poids total. L'autre différence entre UHC et TSP tient au fait que dans le graphe G toute les arêtes ne sont pas présentes. Pour forcer les solutions d'une instance de TSP à ne pas utiliser de telles arêtes, nous allons considérer la matrice de distance contenant une distance 1 pour les paires de sommets de G qui sont bien reliés par une arête et une distance $n+1$ pour les paires de sommets de G qui ne sont pas reliés. Un circuit passant exactement une fois par chaque sommet traversera n arêtes si le graphe a n sommets. S'il ne passe que par des arêtes "autorisées" il sera de poids n, sinon il sera de poids strictement supérieur. Choisir le poids total $k = n$ nous fournit la réduction.

Exercice 14. Systèmes d'équations booléennes.

1. Montrer que déterminer si un système linéaire booléen (c'est-à-dire sur le corps à deux éléments) a une solution est dans P.
2. Que peut-on dire de la complexité de ce même problème pour des équations polynomiales ?

Solution de l'exercice 14.

1. Soit un système (S) de k équations linéaires à n inconnues et à coefficients booléens :

$$(S) \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n = b_k \end{cases}$$

   On cherche à déterminer si (S) admet ou non une solution.

On fait l'algorithme du pivot de Gauss. On remarque que l'on n'a pas besoin de la soustraction ni de la division car on se trouve dans $\{0,1\}$. On élimine une variable x_i à chaque étape de la manière suivante :
— On regarde si une ligne a un coefficient en x_i non nul ;
— On choisit cette ligne comme pivot ;
— On annule le coefficient en x_i des autres lignes par addition, ce qui représente au plus $k(n+1)$ opérations.
On répète ceci au plus n fois pour obtenir un système dont la partie supérieure est triangulaire. Tout ceci a pris un temps polynomial. On peut alors déterminer si ce nouveau système admet ou non une solution en temps polynomial.

2. Notons POL l'ensemble des systèmes polynomiaux satisfaisable. Nous allons montrer que POL et NP-complet.
Tout d'abord, montrons que POL est dans NP. Considérons une instance de ce problème :

$$(S') \begin{cases} P_1(x_1, x_2, \ldots, x_n) = 0 \\ \vdots \\ P_k(x_1, x_2, \ldots, x_n) = 0 \end{cases}$$

Sur la donnée d'un tel système (S') et d'une assignation α des variables $x_1, \ldots, x_n$, on peut déterminer en temps polynomial si α satisfait le système (S').
Nous allons maintenant réduire 3SAT à POL. Soit F sous forme 3CNF : $F = \bigwedge_{i=1}^{m} C_i$ sur les variables $\{x_1, \ldots, x_n\}$. Au littéral ℓ on fait correspondre le polynôme $f(\ell)$ défini par $f(x_i) = y_i$ et $f(\neg x_i) = y_i + 1$. À une clause $C = \ell_1 \vee \ell_2 \vee \ell_3$ on fait correspondre le polynôme $g(C) = (1 + f(\ell_1))(1 + f(\ell_2))(1 + f(\ell_3)) + 1$. Soit $P_i = g(C_i) + 1$ pour $i \in \{1, \ldots, m\}$. Il est immédiat que le système polynomial booléen

$$\begin{cases} P_1(y_1, y_2, \ldots, y_n) = 0 \\ \vdots \\ P_k(y_1, y_2, \ldots, y_n) = 0 \end{cases}$$

a une solution ssi F est satisfaisable. De plus, on peut calculer ce système à partir de F en temps polynomial. On en déduit que POL est NP-dur. En conclusion, POL est NP-complet.

Exercice 15. On rappelle que la classe coNP est la classe des langages L dont le complémentaire $\bar{L}$ appartient à NP.

1. Montrer que $L \in \mathsf{coNP}$ ssi il existe un langage $L' \in P$ et un polynôme p tels que, pour tout $x \in \{0,1\}^*$, $x \in L$ ssi $\forall y \in \{0,1\}^{p(|x|)} (x, y) \in L'$.
2. Montrer que $P \subseteq \mathsf{NP} \cap \mathsf{coNP}$.
3. Soit L_1, L_2 deux langages de $\mathsf{NP} \cap \mathsf{coNP}$. Montrer que leur différence symétrique (l'ensemble des x tels que x appartient à exactement l'un de L_1 ou L_2) est aussi dans $\mathsf{NP} \cap \mathsf{coNP}$.

Solution de l'exercice 15.

1. Il s'agit simplement de la négation de la caractérisation existentielle de NP.
2. On sait que $\mathsf{P} \subseteq \mathsf{NP}$. Donc aussi $\mathsf{P} = \mathsf{coP} \subseteq \mathsf{coNP}$.
3. Si L_1, L_2 sont deux langages de $\mathsf{NP} \cap \mathsf{coNP}$, alors leurs complémentaires aussi. Et toute intersection et union de tels ensemble aussi. Il suffit alors de voir que la différence symétrique de L_1, L_2 s'écrit comme $(L_1 \cap \bar{L_2}) \cup (L_2 \cap \bar{L_1})$.

Exercice 16. Union et intersection de problèmes NP-complets.

1. Soit IND-OR-CLIQUE l'ensemble des couples (G, k) tels que G soit un graphe ayant soit un ensemble indépendant de taille au moins k, soit une clique de taille au moins k. Montrer que IND-OR-CLIQUE est NP-complet.
2. En général, la classe des langages NP-complets est-elle close par union et intersection ?

Solution de l'exercice 16.

1. INDORCLIQUE est bien dans NP : sur la donnée d'un couple (G, k) et d'un ensemble S de sommets, on peut tester en temps polynomial si S est un sous-ensemble indépendant de taille k ou si S est une clique de taille k.
   Pour montrer que INDORCLIQUE est NP-dur on va faire une réduction à partir de IND. Soit un couple (G, k), on lui associe le couple (G', k'), où G' est obtenu en ajoutant m sommets isolés. Cela transforme un sous-ensemble indépendant de taille k en un autre de taille $k + m$, mais si on prend m assez grand le nouveau graphe ne pourra pas avoir de clique de cette taille.
2. Intuitivement, on peut se dire que l'intersection de deux langages pourrait rendre le langage bien plus simple. Plus formellement, si on considère un langage L qui est NP-complet, alors les langages $L_0 = \{0u \in \{0,1\}^* \mid u \in L\}$ et $L_1 = \{1u \in \{0,1\}^* \mid u \in L\}$ sont aussi NP-complets : L_0 est l'ensemble des mots qui commencent par un 0 puis sont un mot de L, la réduction de L à L_0 consiste juste à ajouter un 0 devant le mot. Par contre le langage $L_0 \cap L_1$ est l'ensemble vide, car un mot ne peut pas commencer à la fois par 0 et par 1. Or l'ensemble vide n'est pas NP-complet.
   Une méthode similaire marche pour l'union.